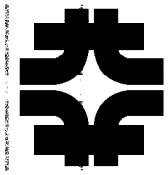


Fermi National Accelerator Laboratory

SVX II Silicon Upgrade

Silicon **R**eadout **C**ontroller

- SRC Users Guide-

26 January, 1998

Harvard University

Participants

J. Huth (PI)
J. Oliver (Eng)
N. Felt (Eng)
S. Bailey (Grad student)
M. Spiropulu (Grad student)
C. Gay (Post doc)
P. Maksimovic (Post doc)
G. Novack (Undergrad)
P. Batra (Undergrad)
N. Scielzo (Undergrad)
S. Harder (Tech)
J. O'Kane (Tech)

1. Introduction

The Silicon Readout Controller (SRC) operates as the interface of the SVX II DAQ subsystem with the Trigger Supervisor (TS) and the CDF Master Clock as well as the upper level controller over the state of the readout electronics. It therefore controls the mediation, interpretation, generation, execution and timing of all the commands that initiate, realize and complete the readout of the silicon detectors with the SVXIII chip set.

1.1. The SRC in the SVX II DAQ Context

Figure 1-1 shows an SRC-centric view of the SVX II DAQ system.

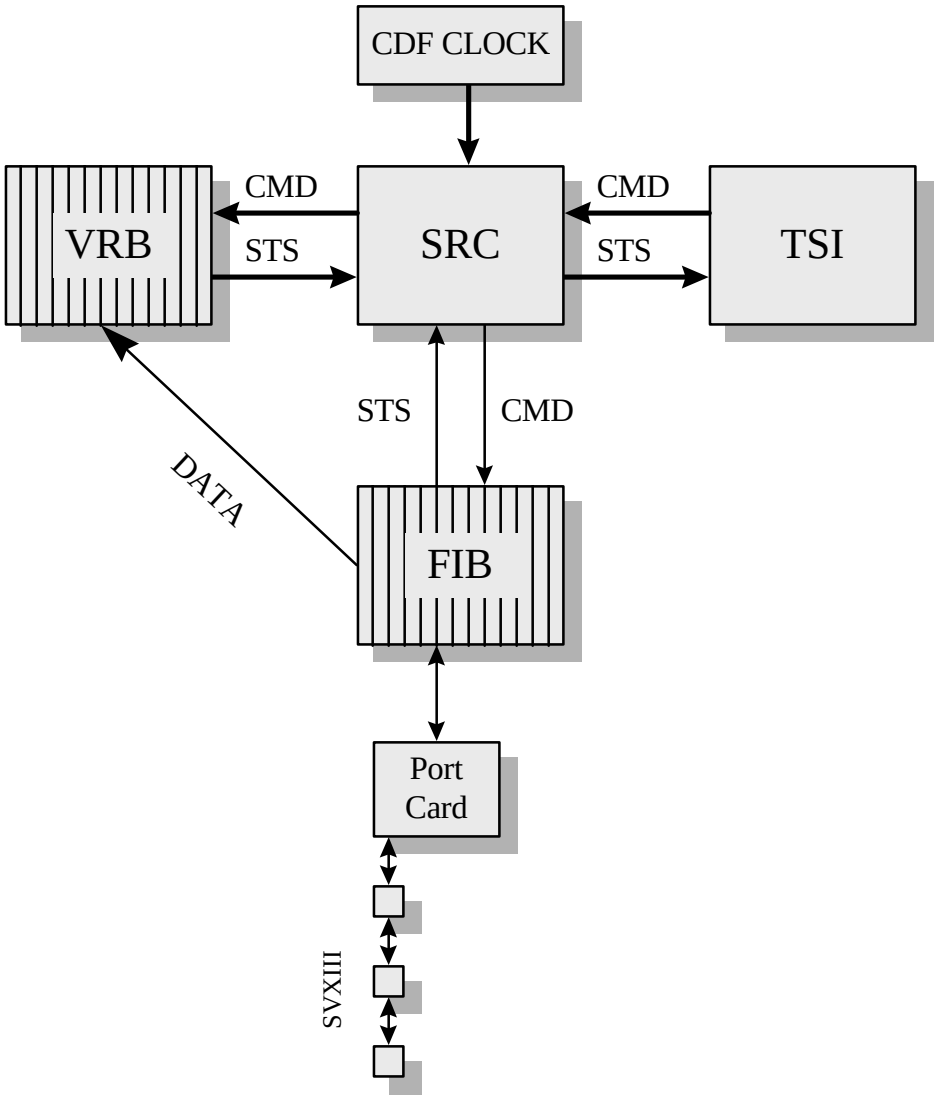


Figure 1-1 - Overview of SVXII DAQ

The SRC receives clock signals and bunch crossing information from the CDF Master Clock. These signals are passed on to the FIB (and thus the SVXIII chips) and are used for the timing of the G-Link connections used throughout the DAQ system.

The Trigger Supervisor Interface (TSI) is the connection between the Trigger Supervisor (TS) and the SRC. The TS sends trigger decisions to the SRC which must coordinate the rest of the SVX II DAQ electronics to achieve the data readout or rejection based upon the trigger decisions. A Level 1 Accept (L1A) from the TS begins the digitization and readout process. This is followed by either a Level 2 Accept (L2A) in which case the data is passed on to level 3, or a Level 2 Reject (L2R) in which case the data readout is aborted. The SRC returns status signals to the TS regarding the state of the data readout.

Upon receipt of a L1A from the TS, the SRC sends signals to the SVXIII chips via the Fiber Interface Boards (FIBs) to mark a given pipeline capacitor and begin the digitization and readout process. Upon completion of the readout process, the SRC frees the previously tagged capacitor storage cell at the next beam gap.

The VME Readout Buffers (VRBs) store data pending a level 2 trigger decision. The SRC manages the buffers and instructs the VRBs on which buffers to fill. The VRBs return status signals indicating when they are busy reading data from the FIBs and when they are passing data on to level 3 after a L2A.

1.2. Internal Sub-Sections of the SRC

Internally the SRC is organized around 7 major components which operate together as a single board level state machine. Each sub-section consists of discrete logic devices and a Xilinx FPGA. In this way, the design can evolve without board level design changes. A overall block diagram is shown in Figure 1-2 followed by a brief discussion of these sub-sections. They are described in more detail in later sections.

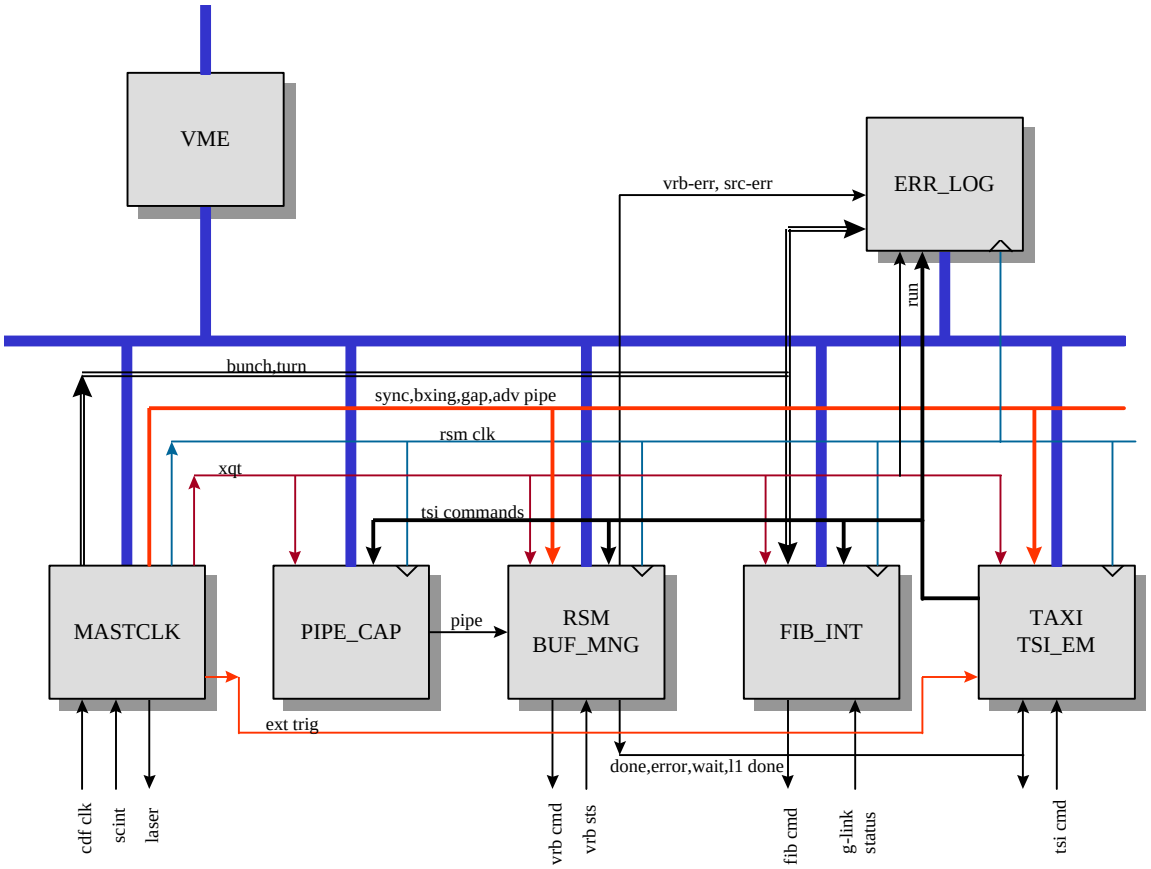


Figure 1-2 - SRC Section Block Diagram

The VME interface implements a standard D32/A32 VME protocol. It decodes addresses of the other SRC sub-sections, determines if access with those sub-sections is legal, and performs simple slave handshake with the VME backplane.

In normal operation, the CDF Master Clock provides the appropriate clock signals for the timing of the SRC. These signals are received by the MASTCLK section and are used to generate SRC board level timing signals, Bunch counters, and Turn counters. A PLL (Phase Locked Loop) low pass filter is provided to remove jitter from the incoming Tevatron RF clock (53.103 MHz) as the G-Links used by the SRC and FIBs require a very stable clock. Programmable emulation modes are available to allow stand alone operation for systems tests as well as cosmic ray and beam test operation.

The function of the PIPE_CAP section is to emulate the address pointer within the SVXIII chips which points to the capacitor sample/hold cell being used for event storage. The result of this emulation becomes part of the data set and is used on-line within the VRBs for purposes of error checking.

The RSM/BUF_MNG (Readout State Machine/Buffer Manager) performs the primary organizational operations of the SRC. It coordinates

the incoming L1A, L2A, and L2R signals from the TSI with appropriate buffer management and instructions to the VRB and FIB in order to read out the data from the SVXIII chips. It also generates status responses to the TSI

The SRC communicates with the FIB crates over a G-Link® fiber optic link. The FIB_INT sub-section initializes the G-Link, monitors it for errors, and coordinates the instructions. There are emulation modes for exercising the FIB as well as history FIFOs for use as a diagnostic (*e.g.* the history FIFO can be displayed like a logic analyzer).

The error logger (ERR_LOG) records error codes returned by the VRB as well as internally generated error codes from the FIB interface and violations of the TSI communications protocol. It contains both error counters for histogramming as well as error recording FIFOs.

The SRC receives signals from the TS via a Transparent Asynchronous Xmitter-Receiver Interface (TAXI) chip set. The SRC must decode these signals, pass them on to the appropriate SRC subsection, and return status signals. Additionally, a TSI emulator is provided which can be loaded with a series of trigger decisions to be sent as if they had been received from the actual TS. This feature allows testing of the SVX II DAQ system at various trigger rates and scenarios. The TSI emulator also generates the trigger accept/reject codes for cosmic ray tests.

2. Trigger Supervisor Protocol

2.1. TSI Commands

The fundamental operations requested by the Trigger Supervisor Interface of the SVX system during normal data acquisition are:

- **L1 Accept** (L1A): An analog storage cell within the SVXIII chip is first tagged to prevent it from being overwritten. Following this, the analog data are digitized and transferred to the VRB's data buffers. The transfer of the digital data is termed a READOUT.
- **L2 Accept** (L2A): Data previously accepted for Level 1 are tagged for transfer to level 3 via the VME backplane in the VRB crate. A buffer so tagged becomes designated as a SCAN buffer.
- **L2 Reject** (L2R): This is a request for data previously accepted at Level 1 to be overwritten and for the corresponding analog storage cell within the SVXIII chip to be made available for additional L1As as soon as possible.

In addition to these commands, the TSI includes a buffer pointer number (0-3) indicating in which of the possible four L1A buffers to store the data. It is the responsibility of the SRC to map these buffer pointer numbers into actual VRB buffer numbers. This will be discussed in greater detail in section 5.4.3.

A complication exists for the SVX system which does not exist for other subsystems. This is due to the nature of the analog storage pipeline used within the SVXIII chip. The SVXIII chip has a requirement that once a storage cell has been tagged, the operation of releasing, or "untagging" it must be done during a clock cycle in which no events take place. This can only be insured if the cell is untagged in a gap in the beam structure. Since the SRC monitors the beam structure in real time, and the TSI does not, the SRC is responsible for determining the correct time to release the cell and reporting this information to the TSI. This necessitates an additional line called **L1_DONE** in the TSI protocol to indicate that a cell has been untagged and is ready to accept new data.

2.2. TSI Status lines

The following lines are used by the SRC to report status to the TSI. All status lines are driven by the SRC to the TSI.

- **L1_DONE**: Asserted for one clock cycle (132ns) indicating that a pipeline capacitor cell has been released.
- **DONE**: De-asserted upon receipt of a L2A. Reasserted on tagging the requested buffer for SCAN. This typically occurs within a few clock cycles as no actual movement of data is required.
- **WAIT**: Asserted by the SRC to indicate that no SCAN buffer is available for L2A.
- **ERROR**: Used to report a fatal error to the TSI.

These lines are used in conjunction with a set of rules governing TSI operation.

2.3. TSI/SRC protocol rules

The TSI/SRC protocol rules follow naturally from a consideration of the operation of the SVXIII chip. The chip has the capability of tagging up to four pipeline capacitor cells at any one time for subsequent readout. Only one of these cells can be digitized and read out at any one time and this must occur in strict time order. When a cell is untagged, it must be during a beam gap so that the cell is not corrupted.

Additionally, the TSI must respect some constraints due to the DAQ system, or in this case, the VRBs. Consistent with CDF DAQ, the VRB has at most four locations in which to store data after a L1A. The VRB has a finite number of SCAN buffers in which it stores data after a L2A prior to its transfer over the VME backplane. The VRB can partition its memory in a variety of ways into buffers. The SVX system allots between four and eight buffers as SCAN buffers.

The rules are as follows:

2.3.1. Rule 1

The number of L1As issued by the TSI, minus the number of L1_DONEs received by it, must never exceed four. This insures that no request for a fifth tagged pipeline capacitor cell is issued.

2.3.2. Rule 2

L2 Accepts will respect a strict handshake with DONE. On receiving a L2A, the SRC de-asserts DONE and re-asserts it once the requested buffer is tagged for SCAN. The TSI cannot issue another L2A until the handshake is complete.

2.3.3. Rule 3

The TSI will not issue a L2A for a pointer until the L1_DONE has been received corresponding to the L1A for the previous pointer.

2.3.4. Rule 4

For every L1A, there will be a L2 decision. Thus, L2 Rejects must be explicit.

2.4. Explicit Timing Examples

The SRC/TSI protocol timing is illustrated at the following examples each accompanied by a timing diagram. Note that times for digitization and readout as well as the times between these cycles are approximate. The top two traces on each diagram show a 132ns clock and the BXing signal which describes the beam structure.

2.4.1. Example 1

Figure 2-1 is an example showing the basic sequence following a L1A by the SRC. After the event is digitized, the pipeline capacitor is replaced in the next available gap, which can be either large (16 missing BXings) or small (2 missing BXings). The readout process may or may not have been started when this gap becomes available. The L1_DONE signal to the TSI occurs at the same time the pipeline capacitor is released. L1_DONE has a duration of 132ns and is always equivalent, in this manner, to the release of the pipeline capacitor in an abort gap.

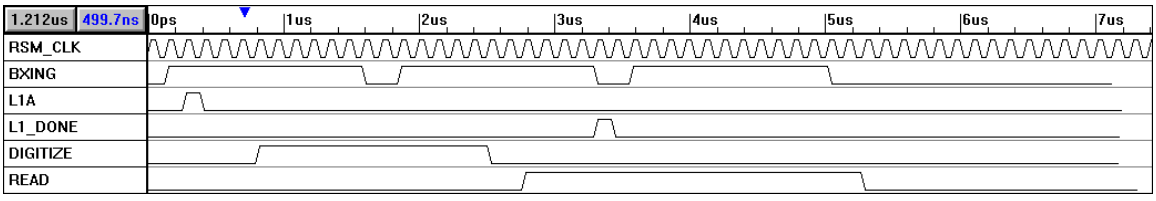


Figure 2-1 - Protocol Timing Example

2.4.2. Example 2

Figure 2-2, shown on a compressed time scale, shows two L1As arriving in rapid succession. The events are digitized and read out and the corresponding L1_DONEs are issued in the available gaps.

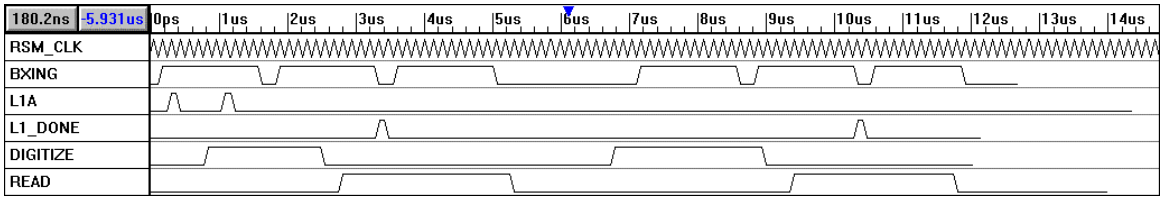


Figure 2-2 - Protocol Timing Example

2.4.3. Example 3

Figure 2-3 is an example of the L2A/DONE handshake. The sequence is uncomplicated since the L2A is issued after the event has already been read out into the VRB, hence the DONE signal is issued immediately, indicating that the corresponding buffer has been designated a SCAN buffer and is awaiting readout to VME.

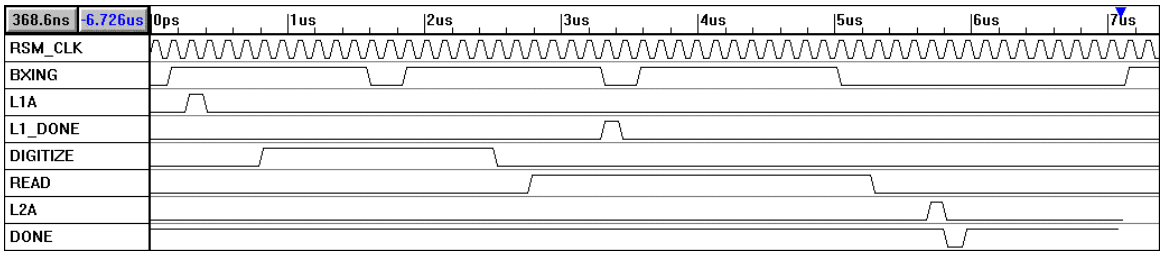


Figure 2-3 - Protocol Timing Example

2.4.4. Example 4

In Figure 2-4, the L2A is issued after L1_DONE but while the event is still being read out to the VRB. DONE is deasserted immediately upon receipt of L2A and is reasserted on completion of readout. This, of course, can only occur when the SVX system is not being used as a component of the L2 trigger.

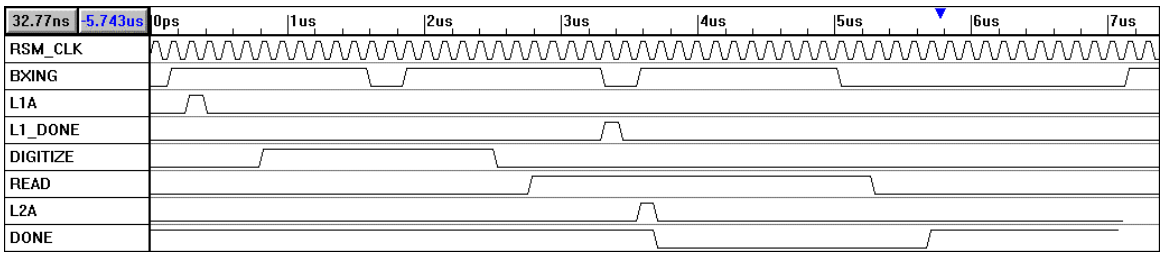


Figure 2-4 - Protocol Timing Example

3. Halt Recover Run

The Halt Recover Run sequence of commands gives the system a way to stop and reset upon a fatal error. After completing the recover command the system will always be in the same known state.

3.1. Error

When the SRC detects an error, either internally or as reported by a Fib Fanout or VRB, it sets the ERROR line to the Trigger Supervisor. The SRC is then allowed to stop the SVX system by putting itself into Halt mode. This will freeze the current error in place and ensure we don't fill the Error FIFO with additional errors which are merely consequences of the first error.

3.2. Halt

The Halt command is the first command of three addressing a fatal error. This command should not modify the status of the system but just stop it from processing any other commands. When a Halt command is issued, the SRC will not accept or process triggers from the TSI. A trigger sent by the TSI will be ignored. The SRC will stop error detection and logging until the end of Recover.

The SRC will try to stop the system in an orderly fashion. If we are in digitize or readout mode the SRC will send an Abort command to the FIB. This will interrupt the current operation in the FIB and cause its completion as quickly as possible. Sending commands from the SRC to the FFO would only be possible with an operational G-link from the SRC to FIB. Since an operational G-link can not be guaranteed at this point, the abort command will not be required. Abort would be the only command that will be sent to the fib. Sync and advance pipe will not be sent to the FIB.

In some cases it will be useful to maintain the state of the DAQ upon a halt command for diagnostics purposes. The SRC will preserve history FIFOs, the buffer manager, the pipeline capacitor emulator, bunch and turn counters and the level queue indicators in their last running state.

3.3. Recover

The entire Initialization and setup will be done in the recover phase as an effort to minimize the processes of the run command. The Recover process can occur under the three following conditions:

- Recover command following a Halt command without an error detected by the SRC. A non SVX error would cause this.
- Recover command following a G-link related error detected by the SRC.
- Recover command following a non G-link related error detected by the SRC.

Upon receiving the command to recover the SRC will have two choices of action depending on whether or not a G-link related error has been detected. If the error is not G-link related in the interest of time they will not be reset. The recover

events following a Halt command without an error detected by the SRC will be the same as those following a non G-link related error.

3.3.1. Error is G-link related

If it has been determined that the error was from the SVX system and G-link related it will be necessary to reset all G-links. When the FFO detects a loss of sync, it will wait for fill frame 0's which will be sent from the SRC to synchronize the connection. The FFO status return cable has four classifications of errors. One classification will signify the status of the G-link and will be used to verify the SRC/FIB Fanout G-link is locked. The SRC will ask the FIB to initialize it's FIB/VRB G-Links by sending the command for fill frame 0 start (h^15) and fill frame 0 stop (h^16). The SRC will send a VRB partial reset (h^0E) command signifying a reset operation along with the data (h^00) signifying the reset is to include all buffers and G-links. This will clear VRB input and output FIFOs, reset VRB GLinks and restart the VRB FPGAs. This takes about 10 msec (mostly waiting for GLinks). The Sync error from the VRB will be used as a closed loop test that all the G-Links are ready. A Sync error from the VRB would indicate that at least one of the VRB receiver G-links is not in Sync.

3.3.2. Error is not G-link related

If it has been determined that the error was from the SVX system and not G-link related, it would not be necessary to reset the G-links. However, the VRB input and output buffers will need be reset with partial VRB reset command. The SRC will send Command (h^0E) signifying a reset operation along with the data (h^01) signifying the reset is to include all buffers but no G-links. This will clear VRB input and output FIFOs and restart the VRB FPGAs.

3.3.3. In all cases

There are certain events which must occur before the system can be started up again, regardless of the source of the Halt or Recover command. TSI events remaining in the SRC queues will be removed. The SRC will clear the buffer manager and Initialize the buffer numbers to the original configuration. The pipeline capacitor emulator will be cleared of all tagged events and the pointers reset. The History and error logger FIFOs will not be cleared of past events. When the number of events grows beyond the FIFO size, new events will simply replace the oldest events in a circular manner.

The SVX pipelines are reset with a command (h^12) sent from the SRC to the FIB. This command only resets the chip pipelines, the serial setup is not redone. It is not known if a partial FIB reset command (h^13) will be needed. The errors detected by the FIB and VRB are latched in their respective modules. This allows VME access to determine more accurately what caused the error. Two commands are to be defined as FIB error reset and VRB error

reset. This will enable the SRC to reset the error flags in the VRB and FIB. This command could be a subset of the FIB partial reset and VRB partial reset. The FIB uses command processing that would allow all or some of the recover commands to be combined into one generic 'Recover' command.

The SRC will use the TSI status line Done as a handshake. If the Recover phase is not complete with a specified amount of time, an error will be flagged. This error would signify a problem that the Halt Recover Run sequence can not recover from.

3.4. Run

The Run command will only enable the system. The SRC will start advancing the pipeline and accept triggers from the TSI.

4. Front panel interfaces

This section describes the pin configurations for the connections between the SRC and the other components of the SVX II DAQ system. For a description of the internal SRC components that interface to these connections, see the appropriate subsections of section 5. A not-to-scale cartoon diagram of the SRC front panel is found in appendix 6.1.

4.1. Master Clock

4.1.1. Cable A – J100

CDF Clock In, Cable A – J100 P/N AMP 3-520459-3 (Key type C)		
Pin #	**Signal Name**	**Comment**
1	CDF_CLK0+	SYNC Input
2	CDF_CLK0-	
3	CDF_CLK1+	BXing Input
4	CDF_CLK1-	
5	CDF_CLK2+	BZero Input
6	CDF_CLK2-	
7	CDF_CLK3+	GAP Input
8	CDF_CLK3-	

Table 4-1 - Master Clock Cable A Bits

4.1.2. Cable B – J101

CDF Clock In, Cable B – J101 P/N AMP 3-520459-3 (Key type C)

Pin #	Signal Name	Comment
1	RF_IN+	53.103 MHz clock input
2	RF_IN-	
3	CDF_CLK4+	Not used
4	CDF_CLK4-	
5	CDF_CLK5+	Not used
6	CDF_CLK5-	
7	CDF_CLK6+	Not used
8	CDF_CLK6-	

Table 4-2 - Master Clock Cable B Bits

4.2. VRB

The SRC is connected to the VRB by a 25 bit copper wire connection carrying both data and commands to the VRB as well as return status and error bits defined by the VRB. These errors are programmable masked as to whether they should be considered fatal (which would initiate a HALT/RESET/RUN sequence) or simply recorded. The bits can be seen in Table 4-3. For a complete pinout of this connection, consult appendix 6.1, Table 6-1.

SRC-VRB Connection – J202		
Bits	**Name**	**Comment**
[7:0]	MSG[7:0]	VRB Data
[11:8]	MSG[11:8]	VRB Command
12	STROBE	VRB Strobe
13	STAT0	READ_BUSY (Reading from FIBs)
14	STAT1	SCAN_BUSY (Writing to level 3)
15	STAT2	VRB_ERROR
[22:16]	STAT[9:3]	VRB_ERR[7:1]
[24:23]	GND	Ground

Table 4-3 - SRC/VRB Bits

The VRB commands sent on bits [11:8] instruct the VRB as to how the data bits MSG[7:0] should be interpreted and what action to take. The SRC buffers these commands and will only send a command when the VRB can process it. The commands are defined in Table 4-4.

Command (in Hex)	Data Means
0	Not used
1	Buffer number to place data coming from FIB
2	Pipe capacitor number – Compare to pipe cap number in data

3	Bunch number – Append to data
4	Scan buffer – Write this buffer to level 3
5	Event ID from TSI – Append to data
6, 7	Not used

Table 4-4 - VRB Commands

4.3. TSI

4.3.1. Command Inputs

The Trigger Supervisor can send the SRC up to two 9-bit words every 132 ns. The control link between the TS and the SRC is implemented by means of the AMD AM7968-175 TAXI (Transparent Asynchronous Xmitter-Receiver Interface) chip set. TAXI chips provide a simple parallel interface through a high speed (up to 175 MHz) serial link while maintaining the data bandwidth required by the system.

There are four types of words send by the TSI to the SRC: Level 1, Level 2, Calibration Enable, and Control Word. These are distinguishable by their first 2 bits and are described in Table 4-5, Table 4-6, Table 4-7 and Table 4-8.

Level 1 Word		
Bits	**Name**	**Description**
[1:0]	C1:C0	Word type; 00 for Level 1 Word
2	L1A	High for Level 1 Accept
3	L1R	High for Level 1 Reject
[5:4]	L1B1:L1B0	Level 1 event buffer address
6	RSRV1	Reserved
[8:7]		Unused

Table 4-5 - TSI Level 1 Command Bits

Level 2 Word		
Bits	**Name**	**Description**
[1:0]	C1:C0	Word type; 10 for Level 2 Word
[3:2]	SESB1:SESB0	Level 2 event buffer address
4	L2A/L2R	High for L2A, low for L2R
[8:5]	EID[3:0]	Event ID

Table 4-6 - TSI Level 2 Command Bits

Calibration Enable Word		
Bits	**Name**	**Description**
[1:0]	C1:C0	Word type; 01 for Calibration Enable Word
[8:2]		Unused

Table 4-7 - TSI Calibration Enable Command Bits

Control Word		
Bits	**Name**	**Description**
[1:0]	C1:C0	Word type; 11 for Control Word
2	HALT	Stop input to level 1
3	RESET	Pipeline reset
4	TEST	Defined for TRACER
5	RUN	Start input to level 1
[8:6]	SCN[2:0]	Scan List bits

Table 4-8 - TSI Control Command Bits

4.3.2. Status Output

The SRC returns four status signals to the TSI which are described in section 2.2 and can be seen in Table 4-9.

SRC-TSI Status Cable – J200 P/N AMP 1-520459-3 (Key type A)	
Pin #	**Signal Name**
1	DONE+
2	DONE-
3	ERROR+
4	ERROR-
5	WAIT+
6	WAIT-
7	L1_DONE+
8	L1_DONE-

Table 4-9 - SRC to TSI Status Return Bits

4.4. FIB

4.4.1. Command Output

The SRC sends a 20-bit word to the FIBs via a 1.5 gigabit/second G-Link and FIB Fanout Every 19 ns. However, with the exception of Sync, data transitions occur with a period of 132ns. The bits are defined in Table 4-10.

Signal	Bits	Definition
GCLK		53 MHz G-Link clock
CMD	[4:0]	5 bit command
XQT	5	Execute Immediate
ADV_PIPE	6	Advance Pipeline
L1A	7	L1 Accept
BUNCH	[15:8]	Bunch Crossing Number

RDQ	[17:16]	10 Read, 01 Digitize, 00 Quiescent (State of DAQ)
PIPE_RD2	18	Return Capacitor to Pipeline
SYNC	19	SYNC(for FIB is PH_01)

Table 4-10 - SRC to FFO G-link Bits

Table 4-11 shows how the command bus codes (CMD) are interpreted by the FIB.

Hex Code	Command
0	No Operation
1	Abort Readout
2	Preamp Stop Reset
4	Preamp Start Reset
5	Readout
6	Digitize
9	Calibration Inject
17	Latch Status
18	Reset SVXIII Chip
19	Reset FIB
20	Reset PC
21	G-Links Send Fill Frames

Table 4-11 - FIB Commands

4.4.2. G-Link Clock Signals

The G-Link output to the FIB operates on ECL voltage level signals. This requires the use of TTL to ECL translators. The data being sent to the FIB uses two adjustable clock delay lines, one for the ECL translators and one for the G-link. All of the FIB data with the exception of Sync is clocked into the translators with a period of 132 ns. This rate provides more than enough setup time for the translators. The Sync data bit is clocked in with a period of 19 ns. To be sure that enough setup time is given, an adjustable clock delay (del 2) is used. The setup time for the translators that are used is 1.5 ns (min) giving a broad range for the clock to be set. It is desired when setting this line for the Sync bit to be clocked in phase with the other FIB data bits while still maintaining the setup requirements. After the ECL translation, the data is then clocked into the G-Link with a period of 19ns. The clock to the G-Link uses an adjustable clock delay (del 3) to insure the setup requirements are met. The G-Link requires a setup time of 6 ns (min).

4.4.3. FIB/G-Link Status Cable

All of the errors detected by the FFO are classified into one of four groups by the FFO. These errors are then returned to the SRC via a transient-protected RS-485 Status Cable receiving status from FIB Fanout modules.. The SRC then logs the errors and decides which of the four classifications should be fatal.

FIB/G-Status Cable – P/N AMP 5-520459-3 (Key type E)		
Pin #	**Signal Name**	**Comment**
1	ERROR A+	to be determined
2	ERROR A -	
3	ERROR B +	
4	ERROR B -	
5	ERROR C +	
6	ERROR C -	
7	ERROR D +	
8	ERROR D -	

Table 4-12 - FIB/G-Link Status Cable Bits

4.5. Indicators

4.5.1. Errors

If the timing lock with the FIB G-Link is lost, the **!GLOCK** indicator is lit. Any fatal error (which initiates a HALT/RESET/RUN sequence) is indicated with the **ERROR** LED.

4.5.2. Geographical Addressing Indicator (GA_IND)

The GA_IND indicator is lit green when VME geographical addressing is in use.

4.5.3. VME ACK/ERR

The VME ACK/ERR indicator flashes green for a legal VME access (acknowledge) and red for an illegal access (error).

4.5.4. Power

Three power indicators are provided:

- **Vcc** : +5 Volts
- **Vee** : -5 Volts
- **Vtt** : -2 Volts

4.5.5. Master Clock

The master clock mode is indicated by a bank of 3 LEDs. The left green LED is for 132 ns mode, the middle orange LED is for 396 ns mode, and the right red LED indicates that the specified mode is locked with the phase locked loop.

4.5.6. Queuing Indicators

Vertical rows of LEDs indicate the number of VRB buffers in use for readout, awaiting a L2 decision, and scanning to level 3. Green indicates that a L1A has been received and readout is in process; orange indicates that the readout to the VRB has been completed and the buffer is awaiting a level 2 decision; red

indicates that a L2A has been received and the scan to level 3 is in progress. Upon receipt of a L2R or completion of the scan to level 3, all lights are cleared.

4.5.7. TSI

The TSI indicator is red for emulation mode, orange for external trigger mode, and green for normal operation.

5. Logical subsections

As previously described in section 1.2, the SRC is divided into seven subsections each of which contains discrete logic devices and a Xilinx FPGA. This section describes each of these in greater detail.

5.1. User Interface

The SRC operates as a board level synchronous state machine at the clock frequency of one seventh of the Tevatron RF (132ns period). Each sub-section operates under the control of one or more internal control registers and a simple mode control state machine. In order to synchronize the operation of the sub-sections, a global execute signal (XQT) is provided. This signal is one clock period in duration and is used by all mode control state machines to initiate the actions specified in the control registers. An attempt is made to unify the meanings of the individual control registers to the extent possible.

5.2. VME Interface

The SRC interface implements an “almost” standard D16/A32 slave interface with geographical addressing. The exception to the literal standard is that single byte data transfers are not permitted in all instances. The reason for this is that word length FIFOs are used which are constructed by pairs of byte wide FIFOs. Allowing byte access, in some cases, invites difficulty in that the upper and lower byte FIFOs could get out of sync. The easy cure is to disallow byte access which, in any case, poses no hardship.

The VME interface FPGA decodes addresses for the various sub-sections, gets permission (or not) from those sub-sections to complete the data transfer, and completes the VME transaction with the appropriate DTACK (or BERR).

5.3. Master Clock

The Master Clock receives timing signals from the CDF Master clock, performs PLL (phase locked loop) filtering on the main timebase of 53.103 MHz, and, from them, derives other board level timing signals. A list of the signals coming from the CDF Master Clock is given in Table 5-1.

5.3.1. CDF Master Clock Signals

Signal	Comment
RF_IN	53.103 MHz symmetric clock synchronized to Tevatron RF

SYNC	19 ns duration pulse, 132ns period
BXing	132 ns duration, indicates beam crossing when qualified by sync
BZ	132 ns duration, indicates “bunch zero” when qualified by sync
GAP	Large “Abort Gap” indicator

Table 5-1 - CDF Master Clock Signals

It is assumed that all incoming signals have edges which are coincident with the rising edge of RF_IN. SYNC is used to indicate the potential p/p-bar crossings in the beam structure which occur in multiples of 7 RF_IN clock periods (132 ns). The remaining signals, BXing, BZ, and GAP indicate the “state” of each potential beam crossing and are qualified by SYNC. They have transitions which are thus coincident with both RF_IN and leading edges of SYNC. *All compensation for setup time requirements of the SRC is performed by adjustments on the SRC so that this need not be done by the CDF Master Clock.*

The expected timing relations of these signals shown in Figure 5-1.

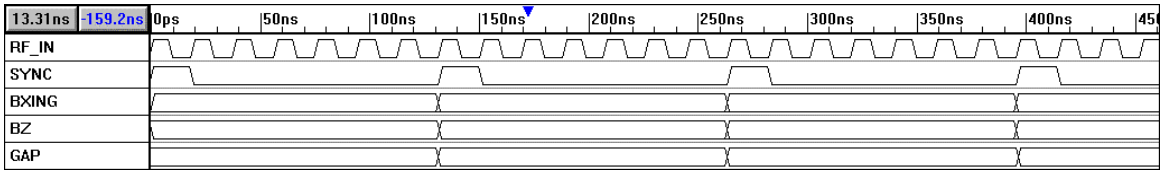


Figure 5-1 - Master Clock Signals Timing

5.3.2. SRC Master Clock Block Diagram

An overall block diagram of the Master Clock subsection is shown in Figure 5-2 followed by a description of its features. For further details (such as the actual state machine diagrams), consult the *SRC Technical Reference Manual*.

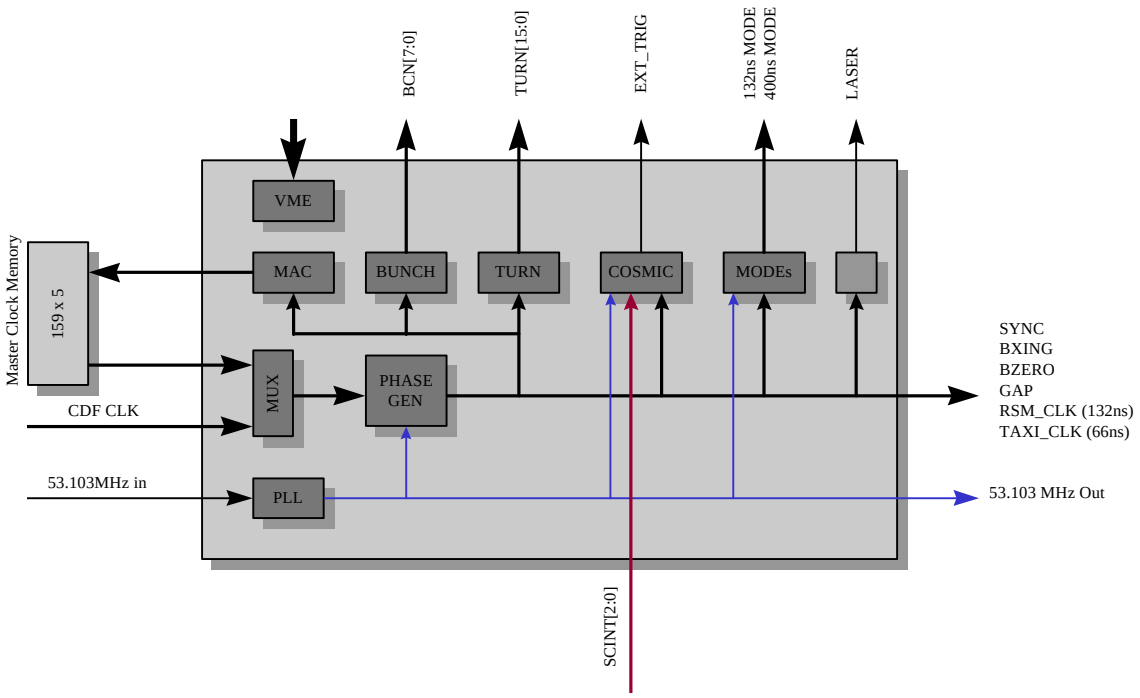


Figure 5-2 - Block diagram of the SRC Master Clock

5.3.3. Phase Locked Loop

The G-Link connections require a very pure clock signal. The SRC master clock interface uses a voltage crystal oscillator which locks within 100 ppm to the central Tevatron frequency of 53.103 MHz. It is very stable and has a low phase noise; the (estimated) RMS of the jitter is less than 25 ps.

5.3.4. Phase Generator

Internal to the SRC master clock FPGA the 53.103 MHz master clock signal is divided into seven phases, PH0 through PH6. This subsection outputs the SYNC, BX, BZero, and GAP signals phase delayed to allow for setup time. It additionally generates the RSM_CLK to clock the readout state machine; the RSM_STROBE which is phased ahead of the RSM_CLK to allow for FIFO reading, setup time, etc.; and the TAXI_CLK used to clock the control frames coming from the trigger supervisor. Figure 5-3 illustrates these signals, including the input from the CDF clock for comparison:

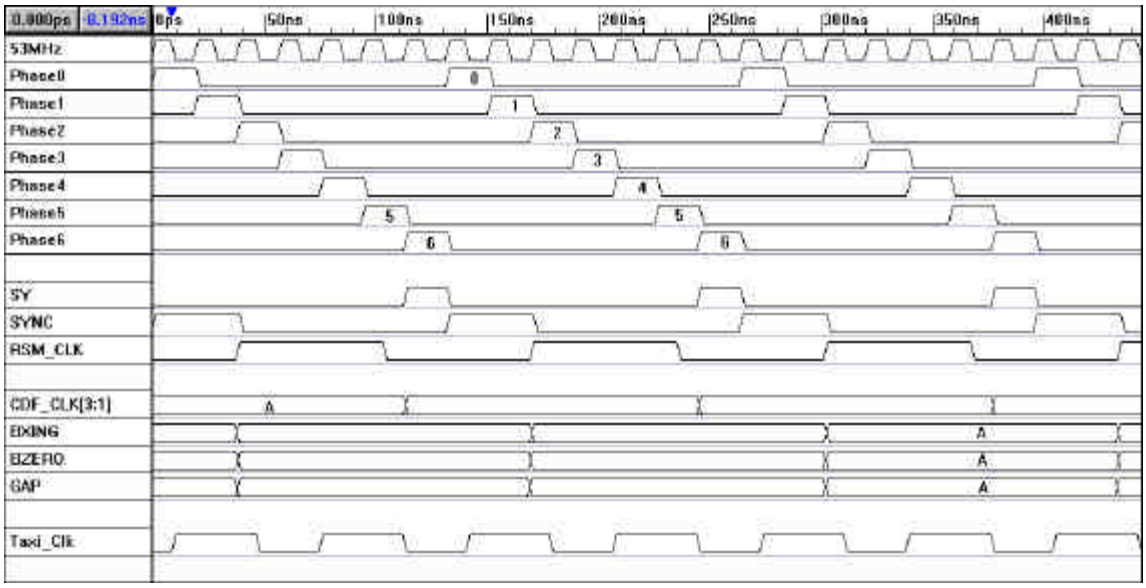


Figure 5-3 - Master Clock Timing

5.3.5. Master Clock Emulator

In normal operation the SRC receives signals from the CDF master clock which it filters and passes on to the rest of the system. For standalone testing (*e.g.* cosmic ray or laser triggers) the master clock may not be available and the SRC must emulate this functionality using its own 53.103 MHz crystal with a fast 8 bit x 159 deep memory. The memory can be loaded with an arbitrary clock signal via the SVX II DAQ software and used to emulate any clock mode (or even a contrived one). This emulated signal is passed to the rest of the system just as the real clock signal would.

5.3.6. Bunch Counter

The bunch counter tracks the system wide bunch crossing number and produces the BUNCH[7:0] signal.

5.3.7. Turn Counter

The turn counter counts the bunch zero's and produces the TURN[15:0] signal.

5.3.8. Cosmic Ray Trigger

The cosmic ray trigger mode has a programmable mask input of three scintillators. It has a programmable 5-bit delay (one RF clock per bit) for the scintillator signal and a programmable width trigger window to allow the cosmic ray triggers to emulate L1As from a real beam crossing. Upon a valid trigger, a EXT_TRIG signal is sent synchronized with the phase delayed BX signal which

corresponds to the input BX during which the scintillator signal was received. Figure 5-4 illustrates these signals.

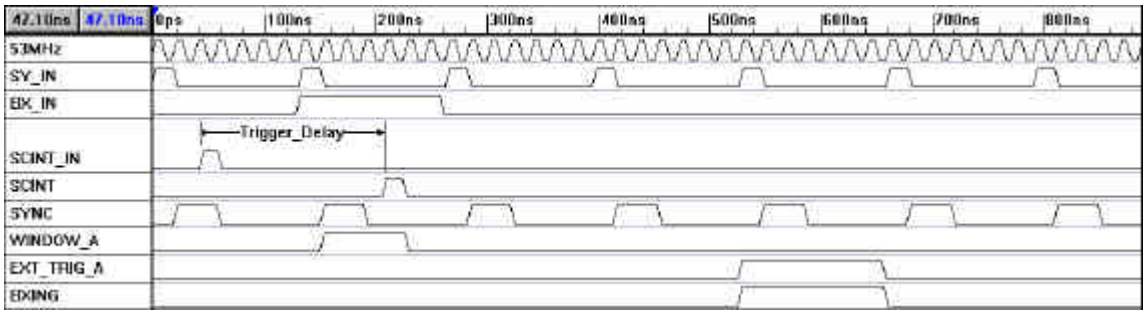


Figure 5-4 - Cosmic Ray Timing

5.3.9. Laser Trigger

In addition to the details of the master clock signals, an arbitrary laser trigger may be programmed. When this signal goes high, the master clock sends a signal to the laser to pulse and instructions to the TSI emulator to proceed with the same trigger accept/reject sequence as for a cosmic ray scintillator trigger. This laser trigger signal is included in the same file as the master clock mode which is sent from the SVX II DAQ software into the master clock memory.

5.4. Readout State Machine and Buffer Manager

5.4.1. General Features

The Readout State Machine (RSM) and Buffer Manager (BUF_MNG) subsection is the core of the SRC. It handles the TSI protocol by processing the incoming L1A/L2A/L2R signals and returning the status signals L1_DONE, DONE, WAIT, and ERROR. The RSM coordinates the readout of the silicon data by generating the appropriate FIB instructions based upon the TSI instructions and the readout status. It also watches the clock for beam gaps and frees up pipeline capacitors when possible. The Buffer Manager portion manages the 4 VRB read buffers and the four to eight VRB scan buffers. Additionally, it queues all instructions to the VRBs such that instructions are not sent until they can be executed. This is accomplished with a two FPGA set, one for the RSM and one for the Buffer Manager.

5.4.2. RSM FPGA

The following is an overview of the readout state machine (Figure 5-5). It is intended to provide a overview of the functionality and the basic purpose of each of the various paths but not to provide a detailed understanding of each of the many states.

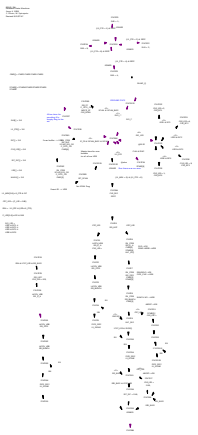


Figure 5-5 - Readout State Machine Diagram

The ground state of the RSM is state 86. Upon receipt of a L1A, the buffer number enters the Level 1 Pointer FIFO for processing and the !L1_PFE (Not Level 1 Pointer FIFO Empty) signal becomes true and the RSM begins the readout process. At state 0 it reads the Level 1 Pointer FIFO (RD_L1PF) and checks if a L2R has been received for this buffer yet. If the L2R has been received already, the RSM proceeds through states 1 through 5 to clear out the buffers, wait for a trigger gap, send a L1_DONE signal, and return to the ground state.

If the L2R has not yet been received at state 0, the RSM begins sending instructions for the readout. To the FIB it sends a commands to begin digitization and readout (which are passed on to the SVX III chips via the port cards). To the VRBs it sends the buffer address (read address) into which to place in the incoming data; the pipeline capacitor number from the pipeline capacitor emulator (which the VRB compares to the number in the data stream); and the bunch number.

At state 9, the RSM begins a counting loop (with INC_DIG and !DIGEQ) to wait for the amount of time it takes for the SVX III chips to digitize the data and begin the readout. If during this time a L2R for this buffer is received, the RSM breaks out of the counting loop (going to state 11); instructs the FIB to abort the readout; clears the buffers; waits for a trigger gap; sends a L1_DONE; and returns to the ground state.

If the digitization counter finishes before a L2R is received, the RSM waits for a trigger gap, sends a L1_DONE, checks again for a L2R and handles it if necessary, and returns to the ground state after readout is completed or aborted. While it is waiting for the readout to be completed (state 15) the RSM watches for a L2R for the next pending L1A and processes it if necessary, thus saving time later. This process is shown on the RSM diagram as the state machine diagram beginning and ending with state 15.

While in the ground state, if a L2R is received (*e.g.* if the above cycle has been completed before the level 2 decision is made), the RSM goes to state 33 to clear the appropriate buffer and return.

If a L2A is received, the appropriate buffer is put into the pending FIFO and !P_FE (Not Pending FIFO Empty) becomes true. Provided that the VRB is not busy scanning another buffer, the RSM begins the loop starting with state 81 which instructs the VRB to begin scanning (writing) the given buffer to level 3. During this process, the SCAN flag is set to high and the VRB sets SCAN_BUSY to high. When the RSM has completed the scanout to level 3, it deasserts SCAN_BUSY and the RSM moves through the loop beginning with state 85 which releases the buffer for future use.

In addition to this main state machine, there is a mini state machine (the one whose ground state is 20) which tracks the buffer which is currently being processed and keeps the SUB variable updated to reflect this.

5.4.3. BUF_MNG FPGA

As seen in the block diagram of the buffer manager given in Figure 5-6. The VRB memory is split into 8 to 12 buffers, each of which can contain the SVX II data for an event. Up to 4 of these can be used for storing data while waiting for a level 2 decision and the remaining 4 to 8 buffers are used as scan buffers while the data is being passed to level 3 upon a L2A. It is the responsibility of the SRC to manage these buffers, indicating which buffer to use to store incoming data; if it should pass that data onto level 3; and free the buffer when no longer needed.

The available buffer numbers are stored in the Empty FIFO and the four pointer registers labeled 0 to 3. When the TSI sends a L1A, it indicates which read buffer pointer (0 to 3) in which to store the data in while awaiting the level 2 decision. The VRB buffer number (0 to 11) which is contained in that read buffer pointer is passed to the VRB, instructing it to store the incoming data in that buffer.

Upon receipt of a L2A, the buffer number in the given buffer pointer is passed to the Pending Register and a new buffer number takes its place, ready for a new L1A. The buffer number is held in the Pending Register until the readout to the VRBs has been completed, whereupon the buffer number is passed to the Pending FIFO. If the readout was already complete at the time of the L2A, the buffer number is passed immediately to the Pending FIFO. In the Pending FIFO, the scan buffers are queued until the VRB can process a scan (write to level 3) request. When this scan has been completed, the buffer number is returned to the Empty FIFO. If a L2R is received for one of the pointers, that buffer number is returned to the Empty FIFO.

In addition to the buffer numbers, the Buffer Manager also passes the event ID (EID) from the TSI to the VRB to append to the data stream. The bunch number and the pipeline capacitor number are also passed to the VRB from the master clock and pipe_cap subsections of the SRC respectively.

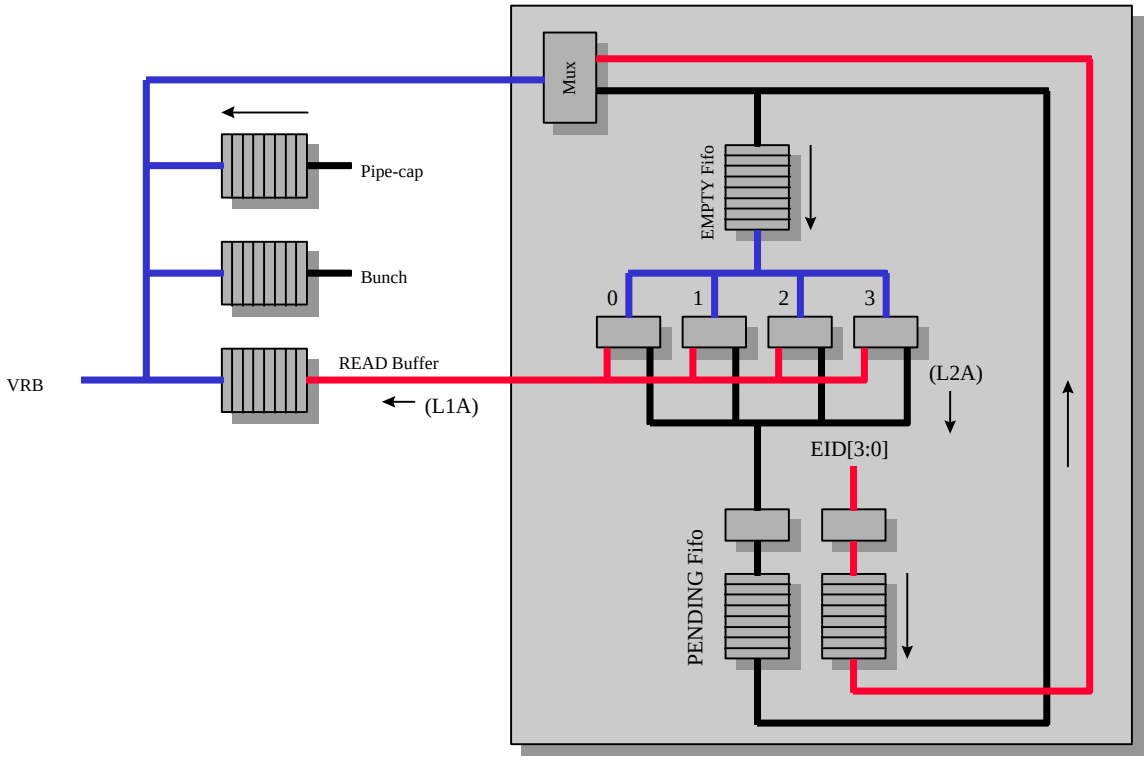


Figure 5-6 - Buffer Manager block diagram.

5.5. FIB Interface

This unit implements the communications with the FIB board by sending the high level commands to the FIB via a G-Link. In addition it implements the G-Link hardware sync; provides an external FIB emulator FIFO and FIB history FIFO (both are 8K words); and receives the status signals of the FIB. The modes and sub-modes can be seen in Table 5-2, Table 5-3, Table 5-4 and Table 5-5.

MODE0 (OFF)		
Sub-mode	**Name**	**Action**
0	Off	Idle
1	Reset History	Reset History FIFO on XQT
2	Reset Emulator	Reset Emulator FIFO on XQT
3-7	Not used	

Table 5-2 - FIB Interface Control Register

MODE1 (MANUAL)		
Sub-mode	**Name**	**Action**
x	Transfer	20-bit FIB control word transfer on XQT

Table 5-3 - FIB Interface Control Register

MODE2 (FIFO EMULATION)

Sub-mode	Name	Action
0	Single Step FIFO	Single step emulator FIFO on XQT
1	Dump FIFO	Dump emulator FIFO on XQT
2-7	Not used	

Table 5-4 - FIB Interface Control Register

MODE3 (NORMAL)		
Sub-mode	**Name**	**Action**
0	FIFO Off	If running turn history FIFO off on XQT
1	FIFO Run	Record on XQT
2	Stop on Error	Record and stop on global error
3-7	Not used	

Table 5-5 - FIB Interface Control Register

5.6. TSI Interface

5.6.1. General Features

The TSI interface must decode two 9 bit words every 132 ns which come from the TAXI link. It provides emulation and history FIFOs for testing and debugging and well as several emulation modes for producing real time L1A, L2A, and L2R signals for system exercises. The TSI interface also processes external triggers (*e.g.* cosmic ray triggers) by generating a given L1A and L2A or L2R signal sequence.

5.6.2. TAXI Decoder FPGA

The TAXI decoder FPGA receives the TAXI signals from the TSI decodes them, and passes the appropriate signals on to the rest of the SRC. The signals are received via an optical link, using the AMD TAXIchip.

5.6.3. TSI Emulator FPGA

In the absence of a trigger supervisor the TSI Emulator will supply the SRC with TSI commands as if they were actually being sent by the trigger supervisor. The Emulator has three distinct modes of operation: Manual, External Trigger, and Emulation.

- In *Manual* mode the L1A, L2R, and L2A commands along with their addresses can be manually selected and sent by the user. This mode allows low rate, step by step approach that aids in the debugging of the entire DAQ system.

- In *External Trigger* mode the trigger signal that is received from the master clock is delayed by a programmable number of virtual beam crossings. After the delay a L1A is given unless it would violate rule #1,

in which case the external trigger would be ignored. As soon as it would be legal a level 2 accept is given. If the level 2 accepts are at a rate faster then can be scanned, the emulator will start rejecting them at level 2.

- The *Emulation* mode allows the emulator to create and send TSI commands at high rates. In this mode the commands that are created are forced to follow the TSI rules. The L2 rate register sets the number of L2Rs for every L2A. There are six Emulation sub-modes which provide functionality for testing various scenarios of the SVX II DAQ system.

    - In *Immediate* sub-mode one L1A is followed immediately by a level one decision. If this decision is a reject, the RSM will not send any commands to the FIB or VRB since the level one accept is rejected Immediately at level 2.
    - In *During Digitize* sub-mode one L1A is given. When the DAQ is in digitize mode a level 2 decision is made. If this decision is a reject an abort command is sent by the RSM to the FIB.
    - In *During Readout* sub-mode one L1A is given. When the DAQ is in readout mode a level 2 decision is made. If this decision is a reject an abort command is sent by the RSM to the FIB.
    - In *After Readout* sub-mode one L1A is given. A level 2 decision is not made until the readout is complete.

    - *Random* sub-mode of operation provides the most realistic TSI emulation. In this mode commands are sent at a random rate with the probability set using registers.
    - The *State Path* sub-mode gives the user the option to create a specific path for the commands to follow. This is useful when a certain sequence of events needs to be studied. The value of this mode is limited by the fact that the FPGA must be recompiled for each new path, *i.e.*, the specific sequence cannot be downloaded via the SVX II DAQ software.

5.7. Pipeline Capacitor Emulator

The Pipeline Capacitor (PIPE_CAP) emulator subsection uses the pipeline capacitor algorithm to determine what capacitor in the pipeline is tagged upon a L1A. Its inputs are the time base from the Master Clock, the L1A signal from the TAXI interface and the L1_DONE from the RSM. Its output is the 6 bit pipeline capacitor number.

The major components of the Pipe Cap can be seen in Figure 5-7. The Pipe Cap emulator contains one memory cell for every capacitor in the pipeline to mark level 1 accepted events. The write pointer decides which capacitor data is being written. The capacitor to be written to is decided by picking the next unmarked cell in the pipeline using skip logic. The L1A pointer points to the capacitor that would be tagged if a level 1 accept is given. The L1A pointer

follows the write pointer by a set number of untagged cells. This latency is achieved using a programmable depth FIFO. When a capacitor is ready to be returned to the pipeline, a Pipe RD2 signal is received from the RSM which frees that cell. The next capacitor to be returned to the pipeline is pointed to by the Pipe RD2 pointer.

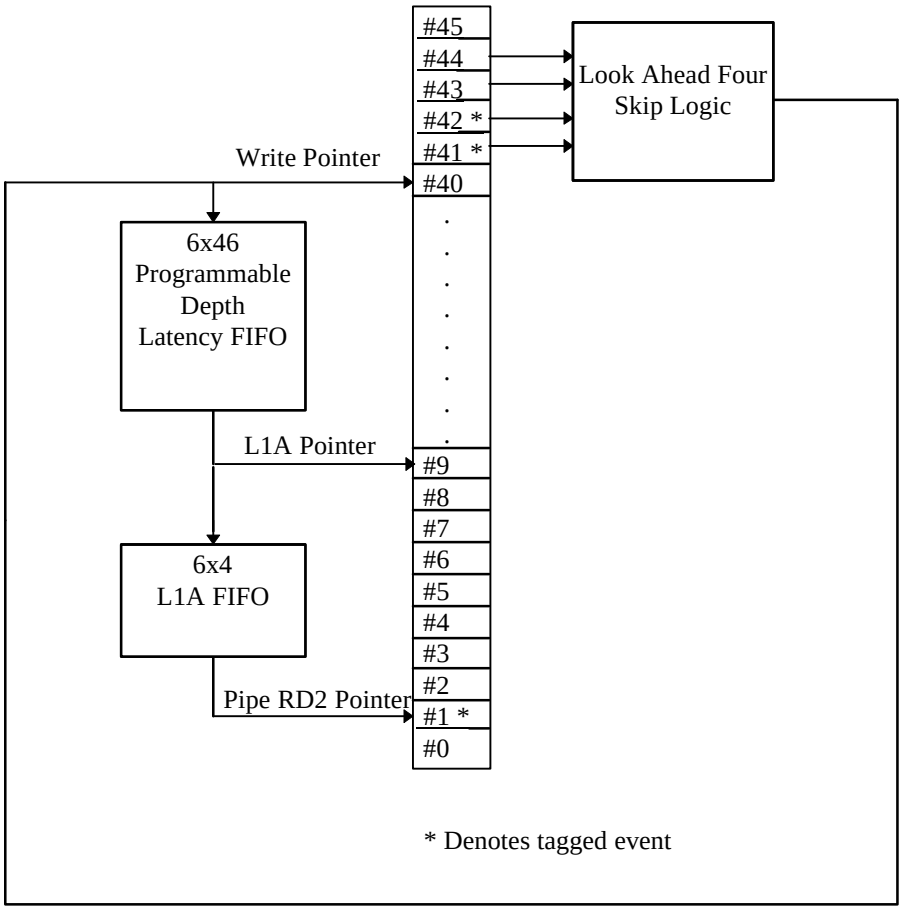


Figure 5-7 - Block diagram of the pipeline capacitor emulator

An example of the Pipe Cap operation can be seen in Figure 5-7. As the pipeline advances, the current write pointer at cell 40 will fall into the programmable depth latency FIFO. This assures the L1A pointer will follow the exact path of the write pointer. The skip logic will look at the four cells ahead of the write pointer and place the write pointer at cell 43. The L1A pointer will be placed at the next position given by the latency FIFO. If a level 1 accept had been given, cell 9 would be tagged and that pointer would fall into the L1A FIFO. If a Pipe RD2 had been given, cell 1 would be untagged and the Pipe RD2 pointer would be placed at the next position given by the L1A FIFO.

5.8. Error Logger

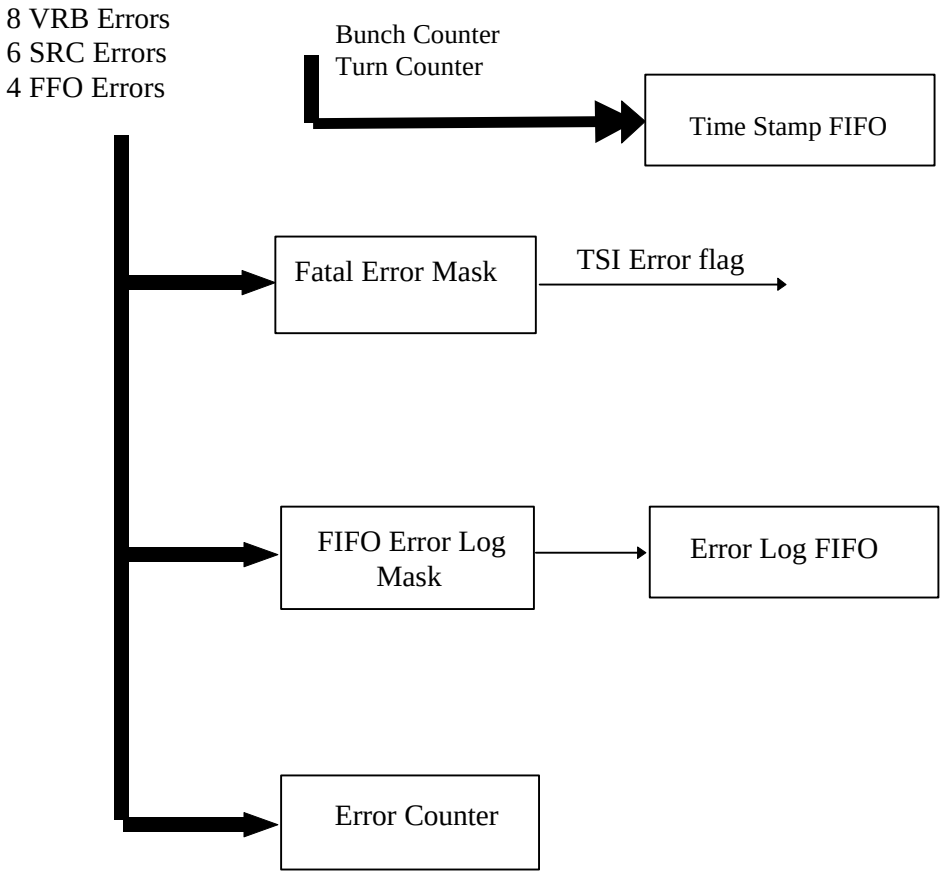


Figure 5-8 - Block diagram of the error logger subsection

There are 6 defined and 2 spare VRB errors returned to the SRC via the VRB status bus (STAT[9:2]). The SRC internally detects 6 errors from violations to the SRC/TSI communication protocol. The FFO maps the errors it detects into 4 four classifications of errors. The 4 classifications are sent to the SRC via the SRC/FFO status return cable. Each of the errors has bunch and turn counter time stamps. There are five 8K deep (by 9 bits each) Error History FIFOs to accommodate the TURN (16 bits), BUNCH (8 bits), VRB_ERRORS (8 bits), FIB Fanout_ERRORS (4 bits), and SRC_ERRORS (6 bits). The 6 SRC_ERROR bits are described in Table 5-6.

For error histogramming there are eighteen (one for each VRB, FFO and SRC error bit) 13 bit counters plus overflow bit. Each individual error can be masked off to be non fatal or masked to be not logged. There will be 4 FIB error lines via a J3 backplane connector. These errors will be counted, logged and included in the fatal error mask. A fatal error will be generator on a large number of non fatal errors using a register for each error.

If a non-fatal error is received, the error line will be cleared by sending a clear error command as soon as possible. Care must be taken determining when it is OK to send this command. If a fatal error is received, the clear error command

will be sent after recover. In the case of any error that error logging will be disabled for that error until it is cleared to prevent multiple counting A block diagram of the Error Logger is given in Figure 5-8. The error logger modes and sub-modes are described in Table 5-7 and Table 5-8.

SRC Error	Error signifies
SRC_ERR0	TSI issues a level 2 accept and there are no scan buffers available
SRC_ERR1	Level 1 accept is received when 4 cells are already active
SRC_ERR2	Level 2 decision is given for a buffer not yet accepted for leve1 1
SRC_ERR3	
SRC_ERR4	Run out of Gap before a stop preamp reset command is given
SRC_ERR5	Is GND, we will never get this error
SRC_ERR6	FIB fatal error
SRC_ERR7	FIB non fatal error

Table 5-6 - SRC Error Bits

:

MODE0 (OFF)		
SUB	**Name**	**Action**
0	OFF	Idle
1	Reset	Reset FIFOs and counters on XQT
2-7	NOT USED	

Table 5-7 - Error Logger Control Register

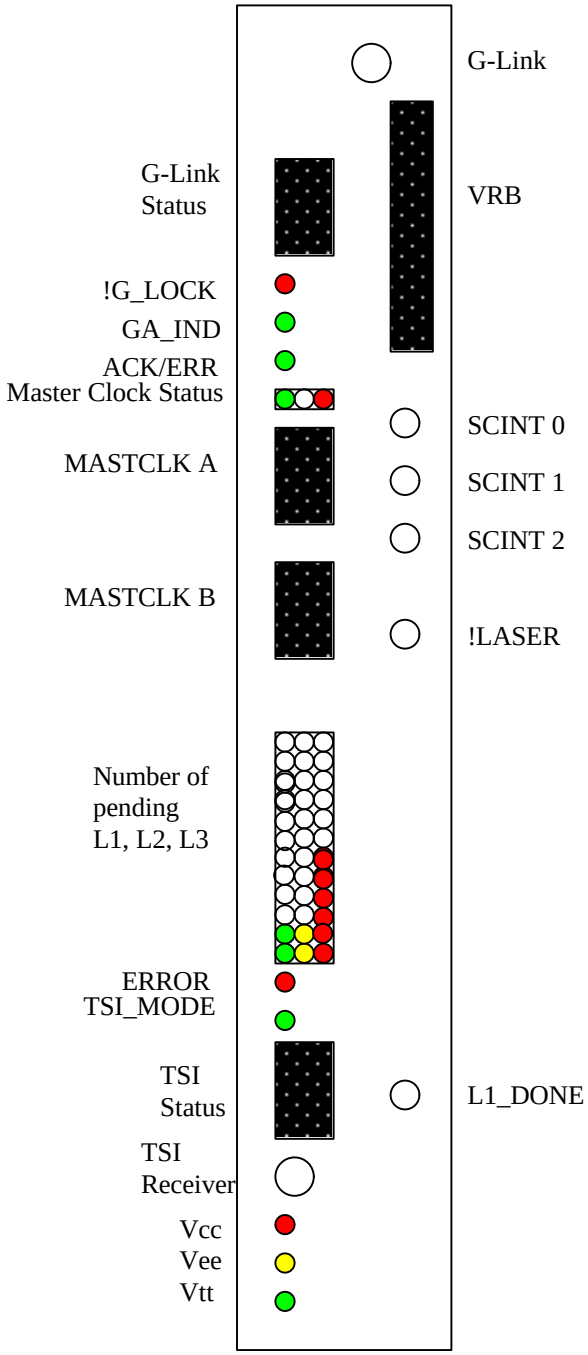
MODE3 (NORMAL)		
SUB	**Name**	**Action**
0	Record	Record ERRORS in RUN(TSI) mode on XQT
1	Record	Record ERRORS on XQT
2-7	NOT USED	

Table 5-8 - Error Logger Control Register

6. Appendices

6.1. Front Panel Diagram

The following is a not-to-scale cartoon diagram of the SRC front panel. Note that this diagram is for the current version of the SRC, not the final version.



SRC-VRB Connection Pin Assignments

SRC-VRB Connection ~ J202 ~ AMP 746789-1		
Pin	Name	Comment
1	Msg 0 -	VRB Data Bi t 0
2	Msg 0 +	
3	Msg 1 -	VRB Data Bi t 1
4	Msg 1 +	
5	Msg 2 -	VRB Data Bi t 2
6	Msg 2 +	
7	Msg 3 -	VRB Data Bi t 3
8	Msg 3 +	
9	Msg 4 -	VRB Data Bi t 4
10	Msg 4 +	
11	Msg 5 -	VRB Data Bi t 5
12	Msg 5 +	
13	Msg 6 -	VRB Data Bi t 6
14	Msg 6 +	
15	Msg 7 -	VRB Data Bi t 7
16	Msg 7 +	
17	Msg 8 -	VRB Command Bi t 0
18	Msg 8 +	
19	Msg 9 -	VRB Command Bi t 1
20	Msg 9 +	
21	Msg 10 -	VRB Command Bi t 2
22	Msg 10 +	
23	Msg 11 -	VRB Command Bi t 3
24	Msg 11 +	
25	Strobe -	VRB Strobe
26	Strobe +	
27	Stat 0 -	READ_BUSY (Reading from FIBs)
28	Stat 0 +	
29	Stat 1 -	SCAN_BUSY (Writing to level 3)
30	Stat 1 +	
31	Stat 2 -	SYNC_ERROR
32	Stat 2 +	
33	Stat 3 -	FRAME_ERROR
34	Stat 3 +	
35	Stat 4 -	IDENTIFIER_ERROR
36	Stat 4 +	
37	Stat 5 -	FORMAT_ERROR
38	Stat 5 +	
39	Stat 6 -	CONTROLLER_ERROR
40	Stat 6 +	
41	Stat 7 -	VRB_ERROR
42	Stat 7 +	
43	Stat 8 -	reserved by VRB
44	Stat 8 +	

45	Stat 9 -	reserved by VRB
46	Stat 9 +	
47	Gnd	Ground
48	Gnd	Ground
49	Gnd	Ground
50	Gnd	Ground

-

Table 6-1 - SRC/VRB Connection Bits

6.2. SRC Addresses

Hex Addr	Name	Bits	Function	Type	Dir	Comments
	VME					
0	Module code		SRC Module code = $01	Byte	R	
2	Revision code		Not used	Byte	R	
4	FPGA Reset register			Byte	R/W	
		7	Not used			
		6	Reset Error Logger			
		5	Reset Pipe Cap Emulator			
		4	Reset TSI Emulator			
		3	Reset FIB Interface			
		2	Reset Readout State Machine			
		1	Reset Master Clock			
		0	Not used			
	MASTCLK					
1000	Master Clock Control Register			Byte	R/W	
		[7:5]	Not used			
		4	Enable Laser			Enable pulses to external laser (or scope trigger)
		3	Clear Turn Counter			

		2	Clear Bunch Counter			
		1	Enable Emulator			
		0	Enable Master Clock			
1002	Master Clock Status			Byte	R	
		[7:3]	Not used			
		2	MODE400			400ns Mode detected
		1	MODE132			132ns Mode detected
		0	LOCK			Locked to external CDF clock
1004	Scintillator Mask Register	[2:0]	Scintillator trigger required	Byte	R/W	Disabled when 000
1006	External Trig Delay	[4:0]		Byte	R/W	External Trigger Window delay in RF clocks
1008	External Trig Width	[4:0]		Byte	R/W	External Trigger Window width in RF clocks
1010	XQT	-	Global Execute Generator	-	W	Generates XQT (Data irrelevant)
1100-123C	Master Clock Memory			Byte	R/W	Beam structure emulation
		4	Laser			Pulse external laser or scope trigger
		3	Gap			Tev large abort gap
		2	BZero			Bunch Zero
		1	BXing			Bunch Crossing
		0	Sync			Sync (always=1)
	RSM					

2000	RSM Control Register			Byte	R/W	
2002	RSM Status Register			Byte	R	
2004	VRB Fatal Error Mask			Byte	R/W	
2006	VRB Buffer Count Reg			Byte	R/W	
2008	SVXIII history depth			Byte	R/W	
200A	Digitize counter/timer			Byte	R/W	
2010	Buffer Pointer 0			Byte	R	
2012	Buffer Pointer 1			Byte	R	
2014	Buffer Pointer 2			Byte	R	
2016	Buffer Pointer 3			Byte	R	
2018	EMPTY Fifo			Byte	R	
201A	PENDING Fifo			Byte	R	
2020	Preamp Reset counter/timer			Byte	R/W	
2022	SVXIII Reset counter/timer			Byte	R/W	
2024	Portcard Reset counter/timer			Byte	R/W	
2026	FIB Reset counter/timer			Byte	R/W	
	FIB_INT					
3000	Control Reg			Byte	R/W	
		[7:4]	Mode[3:0]			See mode descriptions
		[3:0]	Sub-Mode[3:0]			
3002	Status Reg			Word	R	
		[15:10]	Not used			
		9	Emulator Fifo1 Full			
		8	Emulator Fifo1 Empty			
		7	Emulator Fifo0 Full			
		6	Emulator Fifo0 Empty			
		5	History Fifo2 Full			
		4	History Fifo2 Empty			
		3	History Fifo1 Full			
		2	History Fifo1 Empty			

		1	History Fifo0 Full			
		0	History Fifo0 Empty			
3004	G-Link Control Reg			Byte	R/W	
		5	Auto			Auto=1/Manual=0 select
		4	G-Link Run / Initialize			SRC to FIB G-Link status (Manual only)
		3	Reset			Reset G-Link (Manual only)
		2	FF			G-Link Fill Frame type (Manual only)
		1	ED			G-Link Enable Data
		0	DAV			G-Link Data Valid (Manual only)
3006	G-Link Status Reg			Byte	R	
		[7:5]	Not used			
		4	SRC/FIB link locked			SRC/FIB closed loop lock
		3	G-Link Tx only locked			Local Tx Transmitter lock
		2	STAT1(Non Fatal Error)			INIT/RUN dual status line from FIB
		1	STAT0(Fatal Error)			INIT/RUN dual status line from FIB
		0	POK			FIB Power OK (Status from FIB)

3008	FIB History Fifo (lower word)			Word	R/W	
300A	FIB History Fifo (upper byte)			Byte	R/W	
300C	Fib Emulator Fifo			Word	R/W	
300E	Not used					
3010	Not used					
3012	Not used					
3014	Not used					
	TSI Emulator					
4000	TSI_Em Control Reg			Byte	R/W	
		[7:4]	Mode			OFF, MANUAL,FIFO_EM, etc.
		[3:0]	Sub-Mode			Details
4002	Status Reg			Byte	R	
		7				
		6				
		5				
		4				
		3	ERROR			Global status line to TSI
		2	WAIT			Global status line to TSI
		1	L1_DONE			Global status line to TSI
		0	DONE			Global status line to TSI
4004				Word	R/W	
4006	Taxi Emulator Fifo			Word	R/W	
4008	L1 Rate			Word	R/W	
400A	L1 Trigger Delay			Byte	R/W	
400C	L2 Rate			Word	R/W	
400E	L2 Wait			Byte	R/W	
4010	TSI Emulator Data			Word	R	
4012	TSI Emulator Manual Low word			Word	R/W	
		[15:14]	L1A_ADD[1:0]			Level 1 Accept pointer
		13	L1A			Level 1 Accept

		[12:9]	EID[3:0]			4-bit Event ID
		[8:6]	CALIB[2:0]			Calibration code
		5	CAL			Calibration command
		4	CONTROL			Control command
		3	HALT			Global HALT command
		2	RECOVER			Global RECOVER command
		1	RUN			Global RUN command
		0	TEST			TEST command
4014	TSI Emulator Manual High byte			Byte	R/W	
		[8:6]	Not used			
		[5:4]	L2R_ADD[1:0]			Level 2 Reject pointer
		3	L2R			Level 2 Reject
		[2:1]	L2A_ADD[1:0]			Level 2 Accept pointer
		0	L2A			Level 2 Accept
	Pipe Cap Emulator					
5000	Pipe Cap Control Register					
	Error Logger					
6000	Control Register			Byte	R/W	
		[7:4]	Mode			
		[3:0]	Sub-Mode			
6002	Status Register			Word	R	
		9	Fifo4 Almost Full			Error FIFO flags

		8	Fifo4 Empty			
		7	Fifo3 Almost Full			
		6	Fifo3 Empty			
		5	Fifo2 Almost Full			
		4	Fifo2 Empty			
		3	Fifo1 Almost Full			
		2	Fifo1 Empty			
		1	Fifo0 Almost Full			
		0	Fifo0 Empty			
6004	Fifo Error Mask register			Word	R/W	Error is recorded on mask satisfied
6006	Error Fifo 1,0			Word	R	
6008						
600A	Error Fifo 2	[7:0]	BCN[7:0]	Byte	R	Bunch Counter time stamp
600C	Error Fifo 4,3	[15:0]	TURN[15:0]	Word	R	Turn counter time stamp
6010	Error Counter 0	[13:0]	SRC_ERR_0	Word	R	
6012	Error Counter 1	[13:0]	SRC_ERR_1	Word	R	
6014	Error Counter 2	[13:0]	SRC_ERR_2	Word	R	
6016	Error Counter 3	[13:0]	SRC_ERR_3	Word	R	
6018	Error Counter 4	[13:0]	SRC_ERR_4	Word	R	
601A	Error Counter 5	[13:0]	SRC_ERR_5	Word	R	
601C	Error Counter 6	[13:0]	SRC_ERR_6	Word	R	
601E	Error Counter 7	[13:0]	SRC_ERR_7	Word	R	
6020	Error Counter 8	[13:0]	VRB_ERR_0	Word	R	
6022	Error Counter 9	[13:0]	VRB_ERR_1	Word	R	
6024	Error Counter 10	[13:0]	VRB_ERR_2	Word	R	
6026	Error Counter 11	[13:0]	VRB_ERR_3	Word	R	
6028	Error Counter 12	[13:0]	VRB_ERR_4	Word	R	
602A	Error Counter 13	[13:0]	VRB_ERR_5	Word	R	
602C	Error Counter 14	[13:0]	VRB_ERR_6	Word	R	
602E	Error Counter 15	[13:0]	VRB_ERR_7	Word	R	

Table 6-2 - Address Space

6.3. Table of figures